Experimental Results in Audio Indexing

S. Dharanipragada, S. Roukos

IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

ABSTRACT

In this paper we describe the IBM Audio-Indexing System and present some experimental results on the performance of the system on an audio indexing task.

1. Introduction

In today's information technology age we encounter large quantities of information, both audio and video, in our daily lives and there is a great need for efficient ways of searching and retrieving relevant information. The goal of an audio-indexing system is to provide the capability of searching and browsing through audio content. The system is formed by integrating information retrieval methods with large vocabulary continuous speech recognition. In this paper we describe the IBM audio indexing system and present some experimental results on a simple audio indexing task.

The simplest way of searching through speech is by locating potential search keys through wordspotting. Wordspotting, however, is computationally expensive and therefore ceases to be practical for large-scale applications. A more efficient method would be to use a combination of speech recognition and state of the art information retrieval techniques. Our audio-indexing system is such a system. A large vocabulary (64K words) continuous speech recognition system is used to to produce time aligned transcripts of the speech. Information retrieval techniques are then employed on these recognized transcripts to identify locations in the text that are relevant to the search request. These locations with time alignments then specify regions of the speech that are relevant for the request.

Several groups have reported experiments on audio and video indexing recently. For example, Brown et. al. 1994 [1], developed a video mail retrieval system using simple keyword spotting using a fixed (35 keywords) keyword set. This approach is however restricted to applications where the database is small and where the query vocabulary can be defined in advance. The problem can be alleviated to some extent by employing a large vocabulary speech recognizer [2, 3, 4]. However, even with a large vocabulary, coverage is still a problem since many words, such as proper nouns, abbreviations etc, that are significant from a retrieval standpoint, are often found missing from the vocabulary. One way to overcome this problem is by complementing the speech recognizer with a wordspotter for the out of vocabulary (OOV) words. This hybrid approach has been used, with reasonable effectiveness, by Jones et al., 1996 [3] for the Video Mail Retrieval System at Cambridge University and James, 1996 [4] for indexing radio news broadcasts. An alternative approach to overcoming the vocabulary coverage problem as proposed by Schauble et al. [5] and Wechsler et al. [6] is to index based on sub-word units. Published results with this approach are however confined only to simulations. All the above methods for detecting OOV words have the drawback of being computationally expensive and hence may not be very useful for large scale applications. We are currently exploring some new techniques for addressing this problem. Results of our experiments will be reported in future.

In this paper we give a description of our experimental Audio Indexing System and present some results on the retrieval performance of the system on an audio-indexing task. We also present results that highlight the effects of speech recognition errors on the retrieval performance.

2. The Evaluation Corpus

Our Audio-Indexing evaluation corpus consists of approximately 20 hours of radio news broadcasts from the Voice of America covering the time period between May to June 1996. Each day only three broadcasts starting at a different hour and spaced roughly 8 hours apart were downloaded from their internet site. This was done to ensure that the broadcasts are not too similar in content and also to ensure that the collection had several different speakers. The entire collection has about 10 main speakers (both male and female anchors) with several more speakers (correspondents, interviewees etc.) contributing short segments. Each broadcast is typically 6 or 10 mins long and begins with a signature announcement followed by the signature music. A typical news bulletin usually consists of several news stories and often includes reports from correspondents over the telephone line and brief interviews with foreign speakers of English.

The entire speech collection is recognized with a large vocabulary speech recognizer to produce transcripts along with time-alignments for each word in the transcripts. Unlike in the standard information retrieval scenario where the text collection is segmented into documents with each document usually discussing a specific topic/story, story segmentation is not automatically available in this application. We, thus, need a scheme to segment the transcripts into stories. One method is to apply standard topic identification schemes to automatically segment the text into topics, however, a more simplistic solution to this problem is to chunk the transcript into overlapping segments of a fixed number of words and treat each chunk as a separate document. We adopt such an approach in our experiment here, with 100 words in each

# queries	53
average length in words	10
average number of relevant documents per query	11

Table 1: Query statistics

# Gaussians	WER (%)	Decoding speed
180,000	25.7	55×real-time
35,000	30.2	30×real-time

Table 2: Performance of the speech recognizer.

document, resulting in 3412 documents in the collection.

2.1. The test collection

Evaluating an information retrieval systems requires search requests, together with assessments of the relevance of each document to each of these requests. The search requests were collected from independent sources such as newspapers and other news broadcasts appearing during the same period of time. This method of collecting search requests is similar to the TREC evaluation and in general they form a better test for the information retrieval system than "known item retrieval", where users are asked to compose queries after reading the documents. We compiled 85 requests in this manner. Judging the relevance of each document for each of these queries is a time-consuming task. Instead, we took the following approach. We ran our information retrieval system on the document collection with each of these search requests and made relevance judgment of only the top 30 ranked documents for each query. We found that only 53 of the 85 requests had any relevant documents, which can be attributed to the small size of the database. We discarded the requests that did not have any relevant documents from our evaluation set. The query statistics are shown in Table 1.

3. System Description

Our current Audio-Indexing system consists of two components: (1) A large vocabulary continuous speech recognition system, and (2) a text-based information retrieval system. Below we give a brief description of these two components.

3.1. Speech Recognition System

The recognition system used here is based on the large vocabulary continuous speech recognition system described in [7, 8, 9]. The system uses acoustic models for sub-phonetic units with context-dependent tying. The instances of context-dependent sub-phone classes are identified by growing a decision tree from the available training data and specifying the terminal nodes of the tree as the relevant instances of these classes. The acoustic feature vectors that characterize the training data at the leaves are modeled by a mixture of Gaussian pdf's, with diagonal covariance matrices. Each leaf of the decision tree is modeled by a 1-state Hidden Markov Model with a self loop and a forward transition. The IBM system expresses the output distributions on the state transitions in terms of the rank of the leaf instead of in terms of the feature vector and the mixture of Gaussian pdf's modeling the training data at the leaf. The rank of a leaf is obtained by computing the log-likelihood of the acoustic vector using the model at each leaf, and then ranking the leaves on the basis of their scores.

The system used here was trained on the WSJ corpus. The decision tree classifying the sub-phonetic units has around 6000 leaves. We built two systems, one which has a maximum of 30 Gaussians modeling each leaf and a smaller system which has a maximum of 6 Gaussians modeling each leaf. Overall, the two systems have around 180,000 and 35,000 Gaussians respectively. For the language model we use a deleted interpolation trigram model which was also trained on the WSJ corpus with a 64K cased vocabulary. The language model has a perplexity of 253.3 [1] on the WSJ test set. The acoustic space is parameterized by 60 dimensional feature vectors which are obtained by performing a Linear Discriminant Analysis on a 9 frame window of 24 dimensional cepstral coefficients vectors.

The performance of the above system was tested on a test set composed of two 10 min VOA broadcasts and the results are shown is Table 2. The decoding speed in the table are based on an IBM RS6000/590 machine. On the WSJ test set the above system has a WER of 11%. The higher error rate on the VOA test set can be attributed to several reasons: (1) the VOA speech has a large proportion of spontaneous speech whereas the WSJ speech is mainly read speech, (2) the VOA speech is of a lower bandwidth (11KHz) than the WSJ speech, and, (3) the language model is not tuned to the VOA corpus.

3.2. Information Retrieval System

An Information Retrieval System typically works in two phases, the *document indexing* phase and *query-document matching* phase. In the document indexing phase each document in the collection is processed to yield a document description, also known as a document-index, which stands in its place during the retrieval. In our system this processing involves part-of-speech tagging of the text, followed by a morphological analysis of the text, followed by removal of function words using a standard *stop word* list. This is in contrast to the simple stemming and filtering used by most of the current systems. Morphological analysis is a form of linguistic signal processing which has great utility in natural language processing. For instance during morphological analysis, among other decompositions, verbs are decomposed into units designating person, tense and mood of the verb plus the root of the verb. Similarly, nouns are decomposed into their roots with (possibly) a tag indicating the plural form. The written request is processed in an identical fashion to yield a *query*. For example, given the request

[1] The language model was trained with the sentence boundary markers and OOV words included, however, they were not included in the perplexity computation.

```
Security arrangements in Hebron involving
international peace-keepers.
```

the following query is obtained after the processing is done.

```
security arrangement Hebron to_involve
international peace-keepers
```

The main feature of our ranking system is a 2-pass approach. In the first pass, given a query, a matching score is computed for each document and the documents are ranked according to this score. The scoring function is simply a weighting scheme that takes into account the number of times each query-term occurs in the document normalized with respect to the length of the document. Normalization is essential to remove the bias towards longer documents. The scoring function also favors terms that are specific to a document and thus rare (and hence more significant) across the documents. We use the following version of the Okapi formula [10], for computing the matching score between a document d and a query q:

$$S(d,q) = \sum_{k=1}^{Q} c_q(q_k) \frac{c_d(q_k)}{0.5 + 1.5\frac{l_d}{\bar{l}} + c_d(q_k)}\, idf(q_k).$$

Here, q_k is the kth term in the query, Q is the number of terms in the query, $c_q(q_k)$ and $c_d(q_k)$ are the counts of the kth term in the query and document respectively, l_d is the length of the document, $\bar{l}$ is the average length of the documents in the collection, and $idf(q_k)$ is the inverse document frequency for the term q_k which is given by:

$$idf(q_k) = \frac{N - n(q_k) + 0.5}{n(q_k) + 0.5},$$

where N is the total number of documents in the query and $n(q_k)$ is the number of documents that contain the term q_k. The inverse document frequency term thus favors terms that are rare among documents.

In the second pass we re-rank the documents by training a probabilistic relevance model for documents, using the top-ranked documents from the first pass as training data.

Retrieval performance is often measured by two measures *precision* and *recall*. Precision is defined as the percentage of the retrieved documents that are relevant to the query and recall is defined as the percentage of the total number of relevant documents that are retrieved. These two measures can be traded off, one for the other. Often a single *average precision* number is computed by first computing the average of the precision at different recall rates for each query, and then by averaging this number across all queries. A more practical measurement, however, is the precision when a fixed number of documents (often small, between 10 and 20) are retrieved. Another commonly used measure is the rank of the highest-ranked relevant document for each query and the percentage of queries that have relevant documents within a given range of the ranked list of retrieved documents.

We evaluated the performance of our system on a small subset and the entire TREC4 document-collection. The results are tabulated in Table 3.

Total number of documents	Avg. Precision
140	83%
175000	29%

Table 3: IR system performance on TREC4

4. Combining Speech recognition with Information retrieval

All the results reported here are based on the speech recognition system with 35,000 Gaussians which had a WER of about 30%. Figure 1 shows the precision vs recall rate for our audio-indexing system, averaged over the 53 queries. The average pecision after the first pass is computed to be 69.92%. With the second pass the average precision increases to 72.83%, which represents a relative increase of about 4.1%.

As described earlier, another way of presenting the retrieval performance is by plotting the precision vs the number of retrieved documents. This is shown in Figure 2. For example, the precision when the top 10 documents are retrieved is 57.92%. With the second pass this improves to 62.26% which represents a 7% relative improvement in performance.

A third method of measuring the retrieval preformance is by the percentage of queries that have relevant documents within a given range of the ranked list of retrieved documents. This is shown in Table 4. We find, for example, that after the first pass, 87% of the queries have at least one relevant document in the top 5 documents and 96% of the queries have at least one relevant document in the top 10 documents.

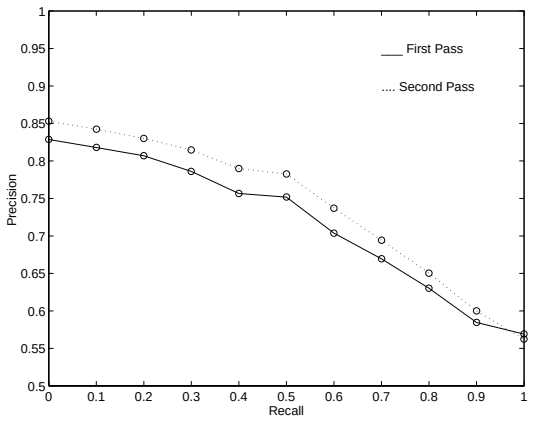

Figure 1: Precision vs Recall rate after the first and second pass.

It is important to evaluate how the performance of the speech recognizer affects the retrieval performance. An obvious way to evaluate this is to run the retrieval experiment on the true text and compare. Unfortunately, we did not have access to the true transcripts for the speech collection used in the ex-

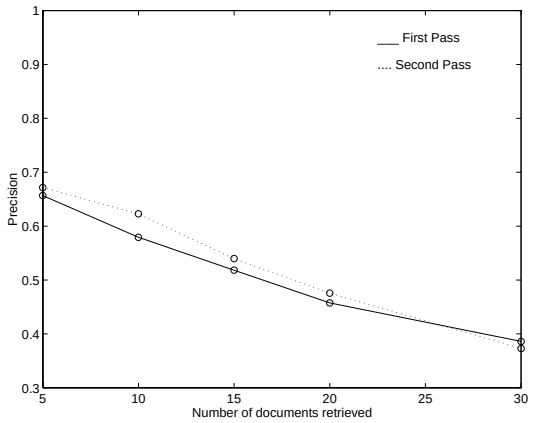

Figure 2: Precision vs number of retrieved documents after the first and second pass.

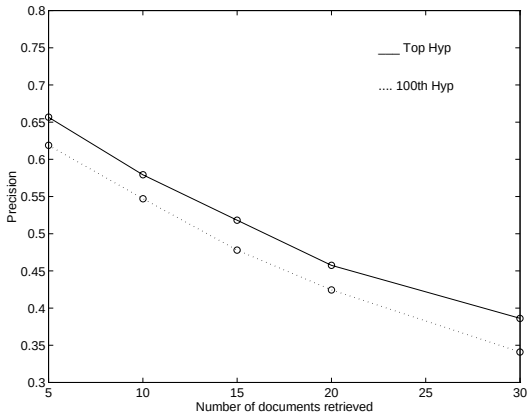

Figure 3: Precision vs number of retrieved documents with the top hypothesis and the 100th hypothesis.

periments here, hence we adopted a different strategy. We produced word lattices for the entire speech collection and used the 100th best hypothesis to simulate a higher word error rate. The 100th hypothesis had a WER of 34.94%. We then repeated the retrieval experiment using documents created using the 100th best hypothesis. The precision vs number of retrieved documents for this case is shown in Figure 3. A comparison of the average precisions for the best and the 100th hypothesis after the first pass shows that the average precision falls from 69.93% to 61.55% and the precision when 10 documents are retrieved falls from 57.92% to 54.72%. Therefore, $\frac{\Delta \text{ avgP}}{\Delta \text{ WER}} = 1.8$, which shows that the retrieval performance is quite sensitive to the performance of the speech recognizer.

5. Conclusions and Future work

We presented an overview of our Audio-Indexing System and reported the performance of our system on an audio-indexing task. Our system has an average precision of about 72% with 96% of the queries having a relevant document in the top 10 ranked list. We also observed that the system is quite sensitive to recognition errors. We are currently exploring new information retrieval methods that are better adapted to the errorful conditions created by the speech recognizer. Current work is also in progress to augment our system with a phone-lattice based scheme for detecting words that are out of the vocabulary of speech recognizer, yielding a open-vocabulary audio-indexing system.

Rank (R)	% queries with at least one relevant document in top R ranks
5	86.79%
10	96.25%
15	98.11%
20	98.11%
30	100 %

Table 4: Rank (R) vs percentage queries with at least one relevant document in the top R ranks after the first pass

References

1. M. Brown, J. Foote, G. Jones, K. Sparc-Jones, and S. Young, "Video mail retrieval by voice: An overview of the cambridge/olivetti retrieval system," in *In Multimedia Data base Management Systems' Workshop at the 2nd ACM International Conference on Multimedia*, 1994.
2. H. Wactlar, T. Kanade, M. Smith, and D. Stevens, "Intelligent access to digital video: The Informedia Project," *IEEE Computer*, vol. 29(5), 1996.
3. G.J.F. Jones and J.T. Foote and G.J.F. Jones and K. Sparc-Jones and S.J. Young, "Robust talker-independent audio document retrieval," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, 1996.
4. D. James, "A system for unrestricted topic retrieval from radio news broadcasts," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, pp. 279–282, 1996.
5. P. Schauble and U. Glavitsch, "Assesing the retrieval effectiveness of a speech retrieval system by simulating recognition errors," *Proc. HLT, ARPA*, pp. 370–372, 1994.
6. M. Wechsler and P. Schauble, "Indexing methods for a speech retrieval system," *Proc. MIRO Workshop, Glasgow*, 1995.
7. L.R. Bahl and P.V. deSouza and P.S. Gopalakrishnan and D. Nahamoo and M.A. Picheny, "Robust methods for context-dependent features and models in a continuous speech recognizer," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, 1994.
8. P.S. Gopalakrishnan and L.R. Bahl and R. Mercer, "A tree search strategy for large vocabulary continuous speech recognition," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, 1995.
9. L. R. Bahl et al ., "Performance of the IBM large vocabulary continuous speech recognition system on the ARPA wall street journal task," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, pp. 41–44, 1995.
10. S.E. Robertson and S. Walker and K. Sparck-Jones and M.M Hancock-Beaulieu and M. Gatford, "Okapi at TREC-3," in *Proc., Third Text Retrieval Conference (NIST special publication)*, 1995.